REFERENCES 79

[80] E. G. Barrantes, D. H. Ackley, S. Forrest, T. S. Palmer, D. Stefanovic, and D. D. Zovi, "Randomized instruction set emulation to disrupt binary code injection attacks," in *Proc. CCS*, pp. 281–289, 2003.

- [81] M. Chew and D. Song, "Mitigating buffer overflows by operating system randomization," Tech. Rep. CS-02-197, Carnegie Mellon University, 2002.
- [82] D. Couroussé, T. Barry, B. Robisson, P. Jaillon, O. Potin, and J.-L. Lanet, "Runtime code polymorphism as a protection against side channel attacks," in IFIP International Conference on Information Security Theory and Practice, pp. 136–152, Springer, 2016.
- [83] M. Jacob, M. H. Jakubowski, P. Naldurg, C. W. N. Saw, and R. Venkatesan, "The superdiversifier: Peephole individualization for software protection," in International Workshop on Security, pp. 100–120, Springer, 2008.
- [84] M. Henry, "Superoptimizer: a look at the smallest program," ACM SIGARCH Computer Architecture News, vol. 15, pp. 122–126, Nov 1987.
- [85] V. Le, M. Afshari, and Z. Su, "Compiler validation via equivalence modulo inputs," in *Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI '14, p. 216–226, 2014.
- [86] B. Churchill, O. Padon, R. Sharma, and A. Aiken, "Semantic program alignment for equivalence checking," in *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI 2019, (New York, NY, USA), p. 1027–1040, Association for Computing Machinery, 2019.
- [87] V. Le, M. Afshari, and Z. Su, "Compiler validation via equivalence modulo inputs," in *Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI '14, p. 216–226, 2014.
- [88] N. Harrand, C. Soto-Valero, M. Monperrus, and B. Baudry, "Java decompiler diversity and its application to meta-decompilation," *Journal of Systems and Software*, vol. 168, p. 110645, 2020.
- [89] M. Zalewski, "American fuzzy lop," 2017.
- [90] K. Zhang, D. Wang, J. Xia, W. Y. Wang, and L. Li, "ALGO: Synthesizing Algorithmic Programs with Generated Oracle Verifiers," arXiv e-prints, p. arXiv:2305.14591, May 2023.
- [91] L. de Moura and N. Bjørner, "Z3: An efficient smt solver," in Tools and Algorithms for the Construction and Analysis of Systems (C. R. Ramakrishnan and J. Rehof, eds.), (Berlin, Heidelberg), pp. 337–340, Springer Berlin Heidelberg, 2008.

80 REFERENCES

[92] P. Kesseli, "Counterexample guided inductive synthesis modulo theories," 2018.

- [93] P. M. Phothilimthana, A. Thakur, R. Bodik, and D. Dhurjati, "Scaling up superoptimization," SIGARCH Comput. Archit. News, vol. 44, p. 297–310, mar 2016.
- [94] R. El-Khalil and A. D. Keromytis, "Hydan: Hiding information in program binaries," in *Information and Communications Security* (J. Lopez, S. Qing, and E. Okamoto, eds.), (Berlin, Heidelberg), pp. 187–199, Springer Berlin Heidelberg, 2004.
- [95] V. Singhal, A. A. Pillai, C. Saumya, M. Kulkarni, and A. Machiry, "Cornucopia: A framework for feedback guided generation of binaries," in Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering, ASE '22, (New York, NY, USA), Association for Computing Machinery, 2023.
- [96] B. Cox, D. Evans, A. Filipi, J. Rowanhill, W. Hu, J. Davidson, J. Knight, A. Nguyen-Tuong, and J. Hiser, "N-variant systems: a secretless framework for security through diversity," in *Proc. of USENIX Security Symposium*, USENIX-SS'06, 2006.
- [97] D. Bruschi, L. Cavallaro, and A. Lanzi, "Diversified process replicae for defeating memory error exploits," in *Proc. of the Int. Performance, Computing, and Communications Conference*, 2007.
- [98] B. Salamat, A. Gal, T. Jackson, K. Manivannan, G. Wagner, and M. Franz, "Stopping buffer overflow attacks at run-time: Simultaneous multi-variant program execution on a multicore processor," tech. rep., Technical Report 07-13, School of Information and Computer Sciences, UCIrvine, 2007.
- [99] L. Davi, C. Liebchen, A.-R. Sadeghi, K. Z. Snow, and F. Monrose, "Isomeron: Code randomization resilient to (just-in-time) return-oriented programming," in NDSS, 2015.
- [100] G. Agosta, A. Barenghi, G. Pelosi, and M. Scandale, "The MEET approach: Securing cryptographic embedded software against side channel attacks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 8, pp. 1320–1333, 2015.
- [101] T. Jackson, B. Salamat, A. Homescu, K. Manivannan, G. Wagner, A. Gal, S. Brunthaler, C. Wimmer, and M. Franz, "Compiler-generated software diversity," in *Moving Target Defense*, pp. 77–98, Springer, 2011.
- [102] A. Amarilli, S. Müller, D. Naccache, D. Page, P. Rauzy, and M. Tunstall, "Can code polymorphism limit information leakage?," in *IFIP International*